PUB-NO: DE004307139A1
DOCUMENT-IDENTIFIER: DE 4307139 A1
TITLE: Serialisation of operating requests in a multiprocessor system
PUBN-DATE: September 8, 1994

INVENTOR-INFORMATION:

NAME	COUNTRY
FISCHER, WOLFGANG DIPL ING	DE
GETZLAFF, KLAUS JOERG DIPL ING	DE
KURZ, BRIGITTE	DE
TAST, HANS-WERNER DIPL ING	DE
WILLE, UDO DIPL ING	DE
WITHELM, BIRGIT	DE

ASSIGNEE-INFORMATION:

NAME	COUNTRY
IBM	US

APPL-NO: DE04307139
APPL-DATE: March 6, 1993

PRIORITY-DATA: DE04307139A (March 6, 1993)

INT-CL (IPC): G06F013/36

EUR-CL (EPC): G06F013/362 , G06F015/16

ABSTRACT:

A method of making processors (PU0 to Pn) of a multiprocessor system (MP) quiescent, or of serialising bus assignments for the individual processors, is described. This serialisation becomes necessary if one or more processors simultaneously want to execute commands which require uninterrupted ownership of the bus for as long as they are being executed, e.g. to prevent damage to the integrity of data.

For this purpose, a quiescent-making network (QN), which connects all processors, is used. Processors which execute such atomic commands receive their bus assignment on a priority basis. The processor with the highest rank puts its competitors into conditional branch command loops (BC loops), in which they wait for a specified condition, e.g. "QN not blocked" (QU+). If this condition is fulfilled, e.g. after the end of an atomic command, the processor with the next lower priority receives the bus.

Processors which are executing other commands are forced into a no-operation state (NOP),

from which they are released after completion of the atomic command of another processor, to resume their interrupted commands.

h c che e e f

(19) BUNDESREPUBLIK DEUTSCHLAND

(12) **Offenlegungsschrift**

(10) **DE 43 07 139 A1**

(51) Int. Cl.[5]: G 06 F 13/36

DEUTSCHES PATENTAMT

(21) Aktenzeichen: P 43 07 139.2
(22) Anmeldetag: 6. 3. 93
(43) Offenlegungstag: 8. 9. 94

(71) Anmelder:
International Business Machines Corp., Armonk, N.Y., US

(74) Vertreter:
Jost, O., Dipl.-Ing., Pat.-Ass., 7000 Stuttgart

(72) Erfinder:
Fischer, Wolfgang, Dipl.-Ing., 7030 Weil, DE; Getzlaff, Klaus Jörg, Dipl.-Ing. (FH), 7036 Schönaich, DE; Kurz, Brigitte, 7030 Böblingen, DE; Tast, Hans-Werner, Dipl.-Ing. (FH), 7039 Weil, DE; Wille, Udo, Dipl.-Ing. (FH), 7038 Holzgerlingen, DE; Withelm, Birgit, 7038 Holzgerlingen, DE

Prüfungsantrag gem. § 44 PatG ist gestellt

(54) Serialisierung von Bedienungsanforderungen in einem Multiprozessor-System

(57) Es wird ein Verfahren zur Ruhigstellung von Prozessoren PU0-Pn eines Multiprozessorsystems MP, bzw. zur Serialisierung von Buszuteilungen für die einzelnen Prozessoren beschrieben, die notwendig wird, wenn einer oder mehrere Prozessoren gleichzeitig Befehle ausführen möchten, die eine ununterbrochene Businhaberschaft für die Dauer ihrer Auführung benötigen, um zum Beispiel Verletzungen der Integrität von Daten zu vermeiden.
Es wird hierzu ein Ruhigstellungsnetz QN verwendet, das alle Prozessoren verbindet. Prozessoren, die solche atomischen Befehle ausführen, erhalten ihre Buszuteilung auf Prioritätsbasis. Der Prozessor mit dem höheren Rang bringt seine Konkurrenten in bedingte Verzweigungsbefehlsschleifen (BC-Schleifen), in denen sie auf eine bestimmte Bedingung warten, z. B. "QN nicht gesperrt" (QU+). Wenn diese Bedingung auftritt, z. B. nach Beendigung eines atomischen Befehls, erhält der Prozessor mit der nächstniederen Priorität den Bus.
Prozessoren, die andere Befehle ausführen, werden in einen No-Operationszustand (NOP) gezwungen, aus der sie nach Beendigung des atomischen Befehls eines anderen Prozessors entlassen werden, um ihre unterbrochenen Befehle weiter auszuführen.

DE 43 07 139 A1

DE 43 07 139 A1

Beschreibung

Prozessoren von Multiprozessorsystemen enthalten Komponenten z. B. Speicher, in die Daten von jedem der Prozessoren eingelesen, oder aus denen Daten von jedem der Prozessoren ausgelesen werden können. Der Befehls- oder Instruktionssatz derartiger Multiprozessoren enthält notwendigerweise auch solche Befehle, für deren Ausführung diese Komponenten für die Dauer eines Befehls einem Prozessor fest zugeordnet sind.

Hier gibt es etwa vier Gruppen von Befehlen, mit denen

1. eine Änderung eines Schlüssels in einem Schlüsselspeicher (Speicher, in dem Schlüssel, beispielsweise für die Zugriffsberechtigung auf bestimmte Speicherbereiche gespeichert sind) bewirkt wird, oder
2. eine Eintragung in allen Adressenübersetzungstabellen aller Prozessoren ungültig gemacht wird (IPTE — Befehl), oder
3. ein Lesen oder Schreiben einer externen Tageszeit-Einheit (Systemuhr), welches eine ungestörte (ununterbrochene) Folge von Befehlen erfordert, durchgeführt wird, oder
4. atomische Lese-/Schreibzugriffe auf den oder die Hauptspeicher oder Caches eines Multiprozessorsystems vorgenommen werden.

In allen in dem vorstehenden Beispiel genannten vier Operationen ist es dringend erforderlich, daß der Prozessor, der eine dieser oben genannten Operationen ausführen möchte solange diese Tätigkeit einstellt, bis alle anderen Prozessoren in ihrem Befehlsablauf eine Stelle erreicht haben, an der sie eine solche Tätigkeit schadlos erlauben können. Diese Stelle ist oft der Beginn eines neuen Befehls oder das Erreichen eines Unterbrechungsfensters in unterbrechbaren Befehlen.

In den meisten Multiprozessorsystemen können von den Prozessoren Befehle der o.g. Art gleichzeitig abgesetzt werden. Unter gleichzeitig soll auch verstanden werden, wenn bei gerade ablaufendem Befehl der eingangs genannten Art eines Prozessors noch ein oder mehrere andere Prozessoren des Multiprozessorsystems solche Befehle ausführen wollen. Beispielsweise dürfen Befehle der Gruppen 1 und 2 gleichzeitig in mehreren Prozessoren ausgeführt werden. So z. B. darf der Prozessor PU1 (Fig. 1) einen IPTE Befehl ausführen, während in der Ausführungsphase in dem Prozessor PU2 ebenfalls ein IPTE Befehl prozessiert wird. Dies führt zu dem Ergebnis, daß beide Prozessoren sich gegenseitig eine (Speicher-)Seite ungültig machen, was aber unschädlich ist, weil beide Befehle keine virtuellen Speicherzugriffe vornehmen.

Befehle der Gruppen 3 und 4, sowie deren Kombinationen mit Befehlen der Gruppen 1 und 2 dürfen dagegen nicht gleichzeitig ausgeführt werden. Die Folgen wären Verletzungen der Integrität der betroffenen Daten, ferner Architekturverletzungen, d. h. es würden Operationen ausgeführt, welche die Regeln der zugrundeliegenden Architektur mißachten und somit nicht erlaubte Ergebnisse produzieren, die zu Mikrocodefehlern und Festfahrsituationen (deadlocks) des Mikrocodeablaufs im Prozessorsystem führen. Diese wirken sich dann leider wie Hardware-deadlocks aus. Es muß also beim Entwurf derartiger Rechnersysteme dafür Sorge getragen werden, daß solche Betriebssituationen vermieden werden.

So ist beispielsweise in der U.S. Patentschrift 5,016,167 beschrieben, wie bei einem Multiprozessorsystem mit einem verschachtelten Speicher das Festfahren verhindert wird, wenn mehrere Prozessoren Zugriffsabsichten auf ein und denselben Teil des Speichers äußern. Es wird hierzu ein Zähler verwendet, der die Anzahl aufeinander folgender Zugriffsanforderungen eines Prozessors zählt, dessen Anforderungen zurückgewiesen wurden. Wenn der Zählerstand einen vorgegebenen Schwellenwert erreicht hat, dann bewirkt ein Überlaufsignal des Zählers eine Sperrung des Zugriffs aller übrigen Prozessoren auf den betreffenden Teil des Speichers solange, bis der bisher nicht erfolgreiche Prozessor mit seinem Speicherzugriff Erfolg hatte. Außerdem wird das Problem der Datenintegrität für Befehle der 3. und 4. Gruppe der o.g. Befehle, sowie daraus resultierende Architekturverletzungen nicht gelöst.

Das Problem bei den bekannten Multiprozessorsystemen ist aber, daß die Strukturen, die ein solches Festfahren und eine Verletzung der Datenintegrität verhindern können, bisher als sehr aufwendige und komplizierte Methoden und Schaltkreise realisiert worden sind, die einen störungsfreien und vor allem schnellen Betrieb und, zudem bei kleineren Systemen, einen wirtschaftlichen Einsatz oft in Frage gestellt haben.

Die Aufgabe der Erfindung besteht somit in der Lösung des Problems, bei Multiprozessorsystemen Festfahrsituationen und insbesondere Verletzungen der Datenintegrität auf ökonomische, sichere und schnelle Weise zu vermeiden.

Gelöst wird diese Aufgabe der Erfindung durch die in den Patentansprüchen angegebenen Merkmale. Vorteilhafte Weiterbildungen und Ausgestaltungen des Gegenstandes der Erfindung sind in den Unteransprüchen dargestellt.

Auf diese Weise wird durch die Erfindung der Vorteil erzielt, daß bei Multiprozessorsystemen bei gleichzeitig abgesetzten Befehlen der eingangs genannten Art mehrerer Prozessoren des Systems Festfahrsituationen sicher, schnell und wirtschaftlich und, im Falle von Befehlen der 3. und 4. Gruppe, sowie deren Kombinationen mit Befehlen der Gruppen 1 und 2, Verletzungen der Datenintegrität vermieden werden.

Im folgenden, wird die Erfindung an durch Zeichnungen erläuterten Ausführungsbeispielen näher beschrieben. Hierbei zeigen:

Fig. 1 ein Blockschaltbild eines Multiprozessorsystems, in das die Erfindung integriert ist,

Fig. 2, 4, 6 Blockschaltbilder einer Schaltungsanordnung, die vorgesehen ist, wenn ein Prozessor alle anderen Prozessoren, die Befehle der 1. bis 4. Gruppe der o.g. Art ausführen wollen, ruhigstellt,

Fig. 3 ein Zeitdiagramm zur Veranschaulichung der Arbeitsweise der Schaltungsanordnung, wenn ein Prozessor, der einen Befehl der 1. bis 4. Gruppe der o.g. Art ausführt, alle übrigen Prozessoren, die sich in der Ausführung normaler Befehle befinden, ruhigstellt und

Fig. 5 ein Zeitdiagramm zur Veranschaulichung der Arbeitsweise der Schaltungsanordnung, wenn mehrere Prozessoren gleichzeitig Befehle der 1. bis 4. Gruppe der o.g. Befehle ausgeben, die jeweils eine Ruhigstellung der anderen Prozessoren erfordern.

Bei dem in Fig. 1 dargestellten Multiprozessorsystem MP sind dessen Prozessoren PU0 bis PUn, sowie ein Speicher-Direktzugriffsadapter DMA und ein Hauptspeicher MS über einen Prozessorbus PB miteinander verbunden, über den Daten, Adressen sowie Komman-

dos, wie "CMD1, CMD2, ... oder CMD Valid" übertragen werden. Ferner ist ein Busanforderungsbus RB und ein Buszuteilungsbus GB vorgesehen, über welche die Prozessoren und der Speicher-Direktzugriffsadapter mit einer Buszuteilungseinheit ARB verbunden sind und über die Busanforderungssignale, sowie Buszuteilungssignale übertragen werden.

Ein Steuerbus CB schließlich stellt eine Steuerverbindung unter den Prozessoren, dem Hauptspeicher und dem Speicher-Direktzugriffsadapter her, über den Steuerinformation ausgetauscht wird.

In Fig. 1 ist ferner noch eine externe Systemuhr TOD, ein Schlüsselspeicher KS, der meist dem Hauptspeicher zugeordnet ist, in dem Schlüssel gespeichert sind, die nur dem Inhaber Zugriffe auf bestimmte Speicherbereiche erlauben, sowie ein Ruhigstellungsnetz QN dargestellt.

Die Hauptaufgabe des Prozessorbusses PB ist, wie schon erwähnt, die Übertragung von Daten, sowie Kommandos und Adressen von und zum Hauptspeicher sowie zwischen den Prozessoren untereinander und dem Speicher-Direktzugriffsadapter.

Die Aufgabe des Speicher-Direktzugriffsadapters besteht in der Steuerung der Kommunikation zwischen der Ein-/ Ausgabeseite, also den Kanälen, an denen die Ein-/Ausgabegeräte angeschlossen sind und dem Hauptspeicher sowie den Prozessoren.

Fig. 3 zeigt in Verbindung mit den Anordnungen in den Fig. 2, 4 und 6, ein Beispiel für einen Ablauf, bei welchem ein Prozessor PU1, der einen Befehl der 2. Gruppe ausführt, alle anderen Prozessoren PU0 und PU2 bis PUn des MP ruhigstellt. Es wird in diesem Beispiel die durchaus reale Annahme gemacht, daß die PUi dieses MP einfache, d. h. nicht sehr komplexe Befehle sehr wohl als festverdrahtete Befehle ausführen können, daß aber komplexere Befehle im Mikro-Ausführungsmodus abgearbeitet werden. Hierzu werden diese Befehle in Form von Mikrobefehlsroutinen interpretiert und ausgeführt.

In der obersten Zeile zeigt Fig. 3 den augenblicklichen Inhalt des laufenden Befehlsregisters CIR dieses Prozessors PU1. Dieser ist ein Befehl IPTE (Invalidate Page Table Entry), mit dem Seitentabellen-Einträge ungültig gemacht werden. Es handelt sich bei diesem Befehl um einen komplexeren Makrobefehl, der vom Prozessor durch eine Folge von Mikrobefehlen interpretiert werden muß.

In der zweiten Zeile in Fig. 3 ist folglich am Inhalt des Mikro-Operationsregisters MOPR(PU1) zu erkennen, daß der Befehl IPTE durch mehrere Mikrobefehle interpretiert wird. Der dritte Mikrobefehl dieser Befehlsfolge in der PU1 ist ein sogenannter ECTLSQ (External Control Set Quiesce)-Befehl, der veranlaßt, daß das Signal PUBR (PB-Anforderung) über den Busanforderungsbus RB zur Buszuteilungseinheit ARB übertragen wird, die PB-Zuteilungen auf der Basis des Prioritätsranges der beteiligten PUs vornimmt.

PU1 wartet nun auf das Buszuteilungssignal PUBG auf dem Buszuteilungsbus GB. Im Mikrozyklus MCYC T3 (Zykluszeit T3) des Befehls ECTLSQ stellt dann PU1 das Kommando CMD1 auf den PB, das folgendes beinhaltet:

1. ein Kommando-Byte ECTL, das "Externe Steuerung für alle PUs" angibt;
2. ein Subkommando-Byte QR, das "Ruhigstellungsanforderung" angibt; und
3. ein Anforderer-Identifizierer RID mit der Nummer der anfordernden PU, im vorliegenden Falle also eine 1 für PU1.

Für die Steuerung der Prozessor-Ruhigstellung oder man kann auch sagen für die Serialisierung bestimmter Abläufe (nämlich die Ausführung der Befehle der eingangs genannten Gruppen 1 bis 4) zwischen den einzelnen Prozessoren, ist in jedem Prozessor PUi eine Schaltungsanordnung vorgesehen, wie sie in den Fig. 2, 4 und 6 für eine MP-Anordnung mit 7 Prozessoren PU0—PU6 dargestellt ist.

Das Kommando CMD1, das PU1 im Mikrozyklus MCYC(T3) auf den Prozessorbus PB gesetzt hat, gelangt in das Prozessorbus-Eingangsregister PBIR aller Prozessoren, außer PU1 und wird 1. im RID Decoder RID DECO decodiert, um den Anforderer (PU1) zu identifizieren. 2. gelangt es auch zum Kommando-Decoder CMD DECO, der das Kommando-Byte ECTL und das Subkommando-Byte QR decodiert. Diese Verhältnisse sind in den Fig. 2 und 3 für PU2 dargestellt.

An seinem Ausgang liegt nach:der Decodierung das Signal QSET, das nach UND-Kombination mit dem in der Verriegelungsschaltung VAL LT verriegelten Signal +CMD VAL, gleichbedeutend mit "Kommando gültig", ein Signal +ECTLSQ ergibt. Dieses Signal, das nicht direkt gleichbedeutend mit dem Mikrobefehl ECTLSQ ist (vgl. Fig. 3, MOPR(PU1)), jedoch von diesem über CMD1 abgeleitet ist, setzt, zusammen mit dem entsprechenden Signal am Ausgang des RID DECO, also am Ausgang 1 (da es PU1 identifiziert), nur die Verriegelungsschaltung QURLT(PU1) in allen Prozessoren, außer PU1. (Das Ausgangssignal der gesetzten QURLT(PU1) bedeutet somit: PU1 hat eine Anforderung "Ruhigstellung, bzw. Serialisierung aller anderen PUs" auf den PB gestellt.) Die Verriegelungsschaltungen QURLT(PU0) und QURLT(PU2) bis QURLT(PU6), für alle anderen PUs bleiben in der Rückstell-Lage.

Da alle PUs dieselbe Schaltungsanordnung nach Fig. 2 aufweisen, findet der oben beschriebene Vorgang in allen PUs des MP statt, wie er für PU2 in Fig. 3 auch an Hand eines Impuls-/Zeit Diagramms dargestellt ist, außer für PU1.

Gleichzeitig mit dem Einstellen der Verriegelungsschaltungen QURLT(PU1) in allen PUs, außer PU1 des MP, stellt PU1 eine Verriegelungsschaltung IQULT(PU1) (interne Ruhigstellung von PU1) ein, die, wie Fig. 4 zeigt, PU — individuell vorgesehen ist, d. h. jede PU hat nur eine eigene solche Verriegelungsschaltung, im Gegensatz zu den QURLTs, von denen jede PU einen vollständigen Satz QURLT(PU0-PU6) aufweist. Das Ausgangssignal BCL einer gesetzten IQULT(PUi) erzeugt über eine ODER-Schaltung GDR eine positive Steuerspannung am Eingang eines Open Drain Teibers DR, der die Funktion einer steuerbaren Impedanz hat. Fig. 3 verdeutlicht den zeitlichen Ablauf der Steuerung auch dieser Verriegelungsschaltungen in dem jeweiligen Prozessor.

Die Einstellung von IQULT(PU1) durch PU1 beendet das Anliegen eines negativen Spannungspegels über den Open Drain Treiber DR an das Ruhigstellungsnetz QN. Mit einer positiven Steuerspannung an seinem Eingang befindet sich nun dieser Treiber im Zustand hoher Impedanz Z (Sperrzustand). Der Spannungspegel (QU) auf dem QN bleibt aber infolge des negativen Ausgangspegels eines sich im Zustand niedriger Impedanz befindlichen (leitenden) DR einer oder mehrerer anderer Prozessoren noch solange negativ, bis sich alle Treiber im Zustand hoher Impedanz (Sperrzustand) befin-

den. Erst dann, wenn alle DR gesperrt sind, geht das Potential auf dem QN auf einen positiven Wert Q+, das über R anliegt, wie Fig. 4 zeigt.

Betrachtet man nun wieder Fig. 3, dann läßt sich feststellen, daß PU2 gerade den Befehl B ausführt, der sich im Befehlsregister CIR für den laufenden Befehl befindet. Im ersten Mikrozyklus T0 von B war ein sogenanntes Torsteuersignal GSST (gate soft stop) wirksam, das die Bedeutung "weicher Stop" (soft stop) bzw."potentieller Stop" hat. Dieses Signal wird immer im ersten Mikrozyklus desjenigen Makrobefehls gebildet, der sich gerade im CIR befindet. Das nächste GSST-Signal kommt nach dem 55. Mikrobefehl wieder. Es sei nun angenommen, daß der Befehl B hier unterbrechbar ist. Ein Mikrobefehl ASST (allow soft stop) im Mikro-Operationsregister MOPR(PU1) definiert die unterbrechbare Stelle und schaltet die entsprechende Verriegelungsschaltung SSTLT(PU2) ein, da QURLT(PU1) noch gesetzt ist (vgl. Fig. 2). Die SSTLT(PU1) kann nicht eingestellt werden, da, wie aus Fig. 2 analog für PU1 entnommen werden kann, das Signal +CMD VALID(PU1) in der in PU1 vorhandenen analogen Schaltungsanordnung eine Einstellung von VAL LT, wegen nicht erfüllter UND-Bedingung am UND-Tor GT CMD VAL verhindert, und ferner auch QURLT(PU1) nicht eingestellt ist.

Die Wirkung dieses weichen Stops ist, daß die Mikrobefehlsdecoder aller anderen PUs, d. h. aller außer PU1, auf NOP (= No operation) gesetzt werden (mittels sog. erzwungener Operationen), so daß die entsprechenden PUs einzyklige Mikrobefehle NOP ausführen, in denen sie quasi auf der Stelle treten.

Es wird also, da QURLT(PU1) in PU2 noch immer gesetzt ist, die SSTLT(PU2) in PU2 gesetzt, mit der Folge, daß PU2 seinen Treiber DR auf hohe Impedanz umsteuert, was aus Fig. 4 analog zu den näher dargestellten Verhältnissen für PU0 und PU6 hervorgeht.

Unter der berechtigten Annahme, daß in der Zwischenzeit auch alle anderen, den Prozessorbus PB anfordernden PUs sich im Zustand des weichen Stops befinden, zieht der Widerstand R alle Anschlüsse VTPU0 bis VTPU6 der PUs mit dem QN auf positives Potental QU+, an dem R mit seinem anderen Anschluß anliegt.

Wie Fig. 3 weiter verdeutlicht, befindet sich PU1 in einer durch das Signal +BC bedingten erzwungenen Mikro-Verzweigungsbefehlsschleife (BC-Schleife; BC = branch on condition) so lange, bis die Verzweigungsbedingung durch "QU = QU+" (quiesce all + = true), (vgl. Fig. 3) in Verbindung mit dem noch im Zusammenhang mit Fig. 5 zu erläuternden Signal QUW (PUi) (quiesce wait) in Fig. 4 aufgehoben wird, ein Zustand, der das Signal +NBCam Ausgang einer UND-Schaltung GA erzeugt. Danach, d. h. eine Mikrozykluszeit Ti später, wird dann der Befehl ECIPTE (external control invalidate page table entry) zur Invalidierung der Seitentabellen-Eintragung ausgeführt. Das Kommando CMD2 wird auf den PB gestellt, das folgenden Inhalt hat:

1. ein Kommando-Byte ECTL, das "Externe Steuerung an alle PUs", bedeutet;
2. ein Subkommando "INVTBL", das invalidiere den Seitentabellen-Eintrag bedeutet;
3. ein Adressenfeld mit der zu invalidierenden Tabellenadresse; und
4. die RID, die in diesem Falle undeterminiert ist.

Die Invalidierung des Seitentabellen-Eintrags wird daraufhin in allen PUs durchgeführt. Danach folgt ein Mikrobefehl ECTLRQ. Nach der dritten PB-Anforderung mittels PUBR und der Zuteilungsgewährung mittels PUBG sendet PU1 das CMD3 über den PB mit der Wirkung, daß der CMD DECO das Signal QRES erzeugt, das, weil ein gültiger CMD (+CMD VALID) vorliegt, das Signal −ECTLRQ zur Folge hat, durch welches die IQULT(PU1) in PU1 und die QURLT(PU1)-en in allen anderen PUs zurückgestellt werden. Ferner wird dadurch das Signal QU auf dem QN wieder negativ, und die SSTLT(PU1−PU6) schalten zurück, wodurch wieder die Ausgangslage im MP hergestellt wird.

Eine etwas anderes Verhalten des MP ergibt sich aus einer Betriebssituation, in der zwei oder mehr PUs Befehle gleichzeitig ausgeben, die eine Ruhigstellung bzw. Serialisierung der anderen PUs erforderlich machen

Fig. 5 zeigt die Verhältnisse in einem Zeitdiagramm, in dem PU2 die anderen PUs ruhigstellen/serialisieren möchte. Der laufende Befehl, der sich im CIR von PU2 befindet ist, wie im vorigen Beispiel, IPTE zur Invalidierung eines Seitentabellen-Eintrags. Der Befehl in PU1 (CIR) ist SEXT (STORE EXTERNAL TIMER) zur Speicherung des Standes eines externen Zeitgebers (Timer). Es sei nun angenommen, daß dieser Timer in einer DMA-Einheit realisiert ist. Bei diesem Makrobefehl SEXT erfordert das Lesen oder Schreiben von Zeit-Werten mehr als einen Lesebefehl SE (SENSE) und auch mehr als einen Schreibbefehl STO (STORE). Die mittels der Lesebefehle erhaltenen Daten gelangen über den PB in in einen Zwischenspeicher (nicht dargestellt) in der PU1 und werden von dort mit mehreren Schreibbefehlen STO über den PB in den Hauptspeicher übertragen. Den ruhiggestellten PUs darf es somit nicht erlaubt sein, während dieser Zeit auf den Timer zuzugreifen, weil sie ja den PB benutzen müßten und außerdem falsche Werte erhielten.

Beide, PU1 und PU2, haben ihre Busanforderung PUBR auf den Prozessorbus PB gegeben. PU1 erhielt ihre Zuteilung PUBG zuerst von der Buszuteilungseinheit ARB auf Grund ihrer höheren Priorität und stellt somit ihr Kommando CMD1 auf den PB. Daher wird in PU2 die QURLT(PU1) eingestellt. PU2 wartet im Mikrozyklus T2 auf ihr PUBG-Signal, das ihr den Zugriff auf den PB gewähren wird. Nach erfolgter Zuteilung stellt PU2 ihr Kommando CMD4 auf den PB.

Der einzige Unterschied dieses Kommandos gegenüber dem CMD1 von PU1 besteht in der anderen RID-Angabe, nämlich RID = 2.

Wie schon im ersten Beispiel an Hand der Fig. 3 gezeigt wurde, haben diese Aktionen zur Folge, daß verschiedene Verriegelungsschaltungen eingestellt werden. Im zweiten Beispiel sind dies die IQULT(PU2) in PU2 und QURLT(PU2) in PU0, PU1 und PU3−PU6. Nimmt man wieder an, daß die übrigen PUs sich schon in dem Zustand des weichen Stops befinden, dann geht das QN auf Pluspotential +QU, obwohl SSTLT(PU2) hier nicht eingeschaltet ist (vgl. Fig. 4, wo SSTLT(PU2) ODER-verknüpft ist mit IQULT(PU2), in Analogie zu IQULT(PU0) oder IQULT(PU6)). Der Open Drain Treiber DR in PU2 wird wieder durch das Ausgangssignal von GDR, das diesesmal über den anderen Eingang der ODER-Schaltung, an dem das Signal +SSTLT(PU2) liegt, erzeugt wurde, gesperrt.

Beide Prozessoren befinden sich nun in der bereits erläuterten Mikro-Verzweigungsbefehlsschleife. Die Verzweigungsbedingung +NBC (no branch on condition) darf aber nur für einen Prozessor, nämlich PU1, beendet werden. Dieses wird durch ein Signal QUW(PU2) (quiesce wait) (Ruhigstellung für PU2 war-

ten) erreicht, das, wie Fig. 4 zeigt über einen Inverter zu einer UND-Schaltung GA übertragen wird, deren UND-Bedingung nun wegen des Inverters nicht mehr erfüllt ist. Das hat zur Folge, daß GA über einen Inverter das Ausgangssignal +BC erzeugt, das PU2 in der BC-Schleife beläßt; PU1 hingegen verläßt die BC-Schleife, da die in Fig. 6 dargestellte Prioritätsschaltung, die, wie übrigens alle anderen Schaltungen nach Fig. 2 aus Teilenummern-Gründen in jeder PUi (fast) identisch vorhanden ist, für die ranghöhere PU1 das Signal QUW(PU1) nicht erzeugt. Somit wird der PU1 mit der niedrigeren Ordnungszahl, aber dem höheren Prioritätsrang, die Buszuteilung gewährt.

Die Schaltungsanordnung in Fig. 6 verwendet hierzu einen PU Adressendecoder PUADR DECO, der beispielsweise aus einer 3 hoch 2 Codekombination, die jede PU an seinen Eingängen I1—I3 bereitstellt, die entsprechende PU identifiziert; und mittels einer nachgeschalteten Decodierungslogik, bestehend aus einigen UND-und ODER-Schaltungen, wird dieses QUW(PUi)-Signal erzeugt. Das Signal QUW (PU2) bleibt für PU2 aktiv, so daß diese PU in der Mikro-Verzweigungsbefehlsschleife bleiben muß.

PU1 kann nun über den PB auf den externen Timer zugreifen oder jede andere Mikrobefehlsfolge ausführen, die Atomizität, d. h. eine unterbrechungsfreie Businhaberschaft des zugehörigen Prozessors, im vorliegenden Falle also der PU1, erfordert.

Am Operationsende setzt PU1 die Verriegelungsschaltungen IQULT(PU1) in PU1 und alle QURLT(PU1) in den anderen PUs wieder zurück, und das Potential auf QN wird wieder negativ. Nach dem letzten Mikrobefehl L (last) fährt PU1 mit dem nächsten Makrobefehl A, der sich nun im Befehlsregister CIR für den laufenden Befehl befindet, fort. Dieser bewirkt in seinem ersten Mikrozyclus T0 das Signal GSST, das zusammen mit dem noch immer aktiven Ausgangssignal von QURLT(PU2) die SSTLT(PU1) einschaltet. Damit wird auch das Potential auf dem QN wieder positiv (alle Treiber DR an QN befinden sich im Sperrzustand).

PU2 kann somit die BC-Schleife verlassen (wegen Signal +BNC) und die Ausführung des sich noch im CIR befindenden IPTE-Makrobefehls fortsetzen.

Die in der vorstehenden Beschreibung genannten Verriegelungsschaltungen sind bistabile Kippschaltungen, aufgebaut als sog. Master/Slave Latche M/S, deren Takteingänge aus Gründen der Vereinfachung weggelassen wurden. Sowohl die Einstelleingänge (Setzeingänge), als auch die Rückstelleingänge (Rücksetzeingänge) sind mit UND-und ODER-Schaltungen versehen, um die gewünschten Signalkombinationen zum Einstellen und Rückstellen dieser Schaltungen zu realisieren.

Der Aufbau der BC-Schleifen bzw. des NOP-Zustandes hängt im wesentlichen von der Architektur der verwendeten Prozessoren ab und macht meist Gebrauch von sogenannten +1 Adressenmodifizierern, welche die Adresse im Mikrobefehls-Adressenregister (nicht dargestellt) je Zykluszeit T (nur bei einzyklischen Mikrobefehlen) um jeweils den Wert 1 erhöhen. Zur Schleifenbildung kann dann die Adressenmodifikation unterbunden werden, so daß für die Dauer weiterer Zykluszeiten der Mikorobefehl im Mikro-Operationsregister MOPR verbleibt. Eine Schleife kann dadurch beendet werden, daß im Falle eines bedingten Verzweigungsbefehls BC, bei nicht erfüllter Verzweigungsbedingung, z. B. +NBC in Fig. 4, die Adresse des Mikrobefehls ECIPTE in das MOPR der betreffenden PU gesetzt wird, beispielsweise dadurch, daß die Folgeadresse nach der Aufhebung der Modifizierungssperre, dann bereits die Adresse der ECIPTE im Steuerspeicher (Mikro-Befehlsspeicher) (nicht dargestellt) des betreffenden Prozessors ist.

Patentansprüche

1. Verfahren in einem Multiprozessorsystem (MP) bei dem mehrere Prozessoren (PU0—PUn) und zentrale Komponenten (MS, DMA, ARB) über ein Bussystem (PB, RB, GB, CB) miteinander verbunden sind, zur Serialisierung von Buszuteilungen (PUBG), wenn zwei oder mehrere Prozessoren auf den Prozessorbus (PUB) zugreifen wollen, um Makrobefehle (z. B. ITPE) auszuführen, die einer ununterbrochenen Buszuteilung über mehrere Zyklen bedürfen, **dadurch gekennzeichnet**, daß Makrobefehle der o.g. Art jeweils als eine Folge von Mikrobefehlen ausgeführt werden, bei welcher denjenigen Mikrobefehlen (z. B. ECIPTE), welche die Operationen der auszuführenden Funktion (z. B. Invalidiere Seitentabellen Eintrag) steuern, ein erster Mikrobefehl (ECTLSQ), vorausgeht, der mittels in jedem Prozessor vorgesehenen Schaltungsanordnungen (Fig. 2, 4, 6) über ein separates, alle Prozessoren verbindendes Ruhigstellungsnetz (QN) eine Ruhigstellung bzw. Serialisierung anderer Prozessoren (z. B. PU0, PU2...PUn), als der anfordernden (z. B. U1), einleitet und denen ein zweiter Mikrobefehl (ECTLRQ) nachfolgt, der mittels der genannten Schaltungsanordnungen über das Ruhigstellungsnetz die Ruhigstellung, nach Ausführung der genannten Funktion wieder aufhebt.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß für eine Betriebssituation, in welcher ein Prozessor (z. B. PU1; Fig. 3), der einen Makrobefehl der genannten Art (z. B. IPTE) ausführen möchte, während andere Prozessoren Befehle einer anderen Art ausführen, der Prozessor mit dem höheren Prioritätsrang die Prozessorbuszuteilung erhält und mittels des genannten ersten Mikrobefehls (ECTLSQ) in jedem Prozessor einen Anforderungsidentifizierer (QURLT(PU1)) einstellt, welcher seinerseits, jeweils zu einer bestimmten Taktzeit (z. B. T0) zu Beginn eines Makrobefehls in einem anderen Prozessor oder bei Vorliegen eines Unterbrechungsfensters (ASST) in einem unterbrechbaren Befehl in einem anderen Prozessor, in jedem anderen Prozessor eine erste bistabile Kippschaltung (SSTLT(PU0, PU2—PUn)) einstellt, deren Ausgangssignal (+SSTLT(PU0, PU2-n) im zugehörigen Mikro-Operationsregister (MOPR (PU0, PU2-n)) einen einzykligen NOP-Mikrobefehl erzwingt, bei welchem der Prozessor keine Funktion ausführt, daß der genannte erste Mikrobefehl ferner im eigenen Prozessor (z. B. PU1) zu einer bestimmten Taktzeit (z. B. T4) seines Mikrozyklus eine zweite bistabile Kippschaltung (z. B. IQULT(PU1)) einstellt, die über eine steuerbare Impedanz (DR) an das Ruhigstellungsnetz QN angeschaltet ist, wobei deren Ausgangssignal (BCL) die steuerbare Impedanz in den Zustand hoher Impedanz einstellt, wodurch das Signal (+BC) erzeugt wird, das im zugehörigen Mikro-Operationsregister MOPR(PU1) einen BC-Mikrobefehl erzwingt bei dem der Prozessor auf eine Bedingung (QU=QU+) wartet, daß aber die steuerbaren Impedanzen aller anderen Prozessoren über das Aus-

gangssignal (+SSTLT(PU0, PU2–PUn)) ihrer jeweiligen ersten bistabilen Kippschaltungen in den Zustand hoher Impedanz eingestellt werden, so daß, wenn alle PUs ihre Anschlüsse (VPU0 – VPUn) auf hohe Impedanz geschaltet haben, das Ruhigstellungsnetz über einen Widerstand (R) auf +Potential (QU+) liegt und die genannte Bedingung (QU=QU+) erfüllt ist, wodurch die BC-Mikrobefehlsschleife aufgehoben und der nächste Mikrobefehl (ECIPTE) in das Mikro-Operationsregister des Prozessors zur Durchführung der eigentlichen Funktion gelangt und, daß schließlich nach Beendigung dieses Mikrobefehls der genannte zweite Mikrobefehl (ECTRLQ) zu einer bestimmten Taktzeit (z. B. T4) in seinem Mikrozyklus die zweite bistabile Kippschaltung (z. B. IQULT(PU1)) wieder zurückstellt, so daß die zugehörige steuerbare Impedanz wieder in den leitenden Zustand gebracht und das Potential auf dem Ruhigstellungsnetz wieder negativ wird und ebenso den Anforderungsidentifizierer (z. B. QURLT(PU1)) in allen Prozessoren zurückstellt, so daß die ersten bistabilen Kippschaltungen (SSTLT(PU0, PU2-n)) in allen anderen Prozessoren zurückgestellt werden und diese aus dem NOP-Zustand austreten und mit ihrem nächsten Makro- oder Mikrobefehl fortfahren.

3. Verfahren nach Anspruch 1 , dadurch gekennzeichnet, daß für eine Betriebssituation, in welcher zwei (z. B. PU1 und PU2) oder mehrere Prozessoren Makrobefehle der genannten Art gleichzeitig ausführen möchten, der Prozessor (PU1; Fig. 5) mit dem höheren Prioritätsrang die Buszuteilung erhält und mittels des genannten ersten Mikrobefehls (ECTLSQ) in jedem Prozessor einen Anforderungsidentifizierer (QURLT(PU1)) und und im eigenen Prozessor zu einer bestimmten Taktzeit (z. B. T4) seines Mikrozyklus eine zweite bistabile Kippschaltung (IQLT(PU1)) einstellt, die über eine steuerbare Impedanz (DR) an ein Ruhigstellungsnetz (QN) angeschaltet ist, wobei deren Ausgangssignal (BCL) die steuerbare Impedanz in den Sperrzustand einstellt, wodurch ein Signal (+BC) erzeugt wird, das im zugehörigen Mikro-Operationsregister (MOPR(PU1)) den BC-Befehl (bedingte Verzweigung) erzwingt, bei dem der Prozessor auf eine Bedingung (QU=QU+) wartet, daß ferner nun der Prozessor mit dem nächst niedrigen Prioritätsrang (PU2) die Buszuteilung erhält, so daß deren Anforderungsidentifizierer (QURLT(PU2)) in allen Prozessoren, sowie die zweite bistabile Kippschaltung (IQULT(PU2)) im eigenen Prozessor eingestellt wird, wodurch, wenn alle anderen Prozessoren ihre steuerbaren Impedanzen gesperrt haben, das Potential auf dem Ruhigstellungsnetz auf QU+ geht, obwohl die erste bistabile Kippschaltung (SSTLT(PU2) zurückgesetzt ist, so daß beide anfordernden Prozessoren sich in einer BC-Schleife befinden, daß nun ferner ein Ruhigstellungs-Wartesignal (QUW(PU2)) mittels einer logischen Schaltungsanordnung (Fig. 6) für den (die) rangniederen Prozessor(en) (PU2) erzeugt wird, das die BC-Schleife für diesen Prozessor aufrechterhält, hingegen die Nichterzeugung eines Ruhigstellungs-Wartesignals (QUW(PU1)) beim ranghöheren Prozessor die BC-Schleife aufhebt, wodurch der nächste und die weiteren Mikrobefehle der eine Atomizität erfordernden Mikrobefehlsfolge ausgeführt wird, an deren Ende die zweite bistabile Kippschaltung dieses Prozessors und die Anforderungsidentifizierer dieses Prozessors in alle Prozessoren zurückgestellt werden und-das Ruhigstellungsnetz wieder auf -Potential gebracht wird, und daß schließlich durch den nächsten Makrobefehl (A) dieses Prozessors zur bestimmten Zykluszeit (T0) die erste bistabile Kippschaltung (SSTLT(PU1)) des ranghöheren Prozessors eingestellt und das Ruhigstellungsnetz wieder auf QU+ Potential geht, wodurch die BC-Schleife für den rangniederen Prozessor (PU2) beendet wird, so daß er nun die unterbrochene Makrobefehlsfolge des IPTE Makrobefehls beenden kann.

Hierzu 6 Seite(n) Zeichnungen

Nummer: DE 43 07 139 A1
Int. Cl.[5]: G 06 F 13/36
Offenlegungstag: 8. September 1994

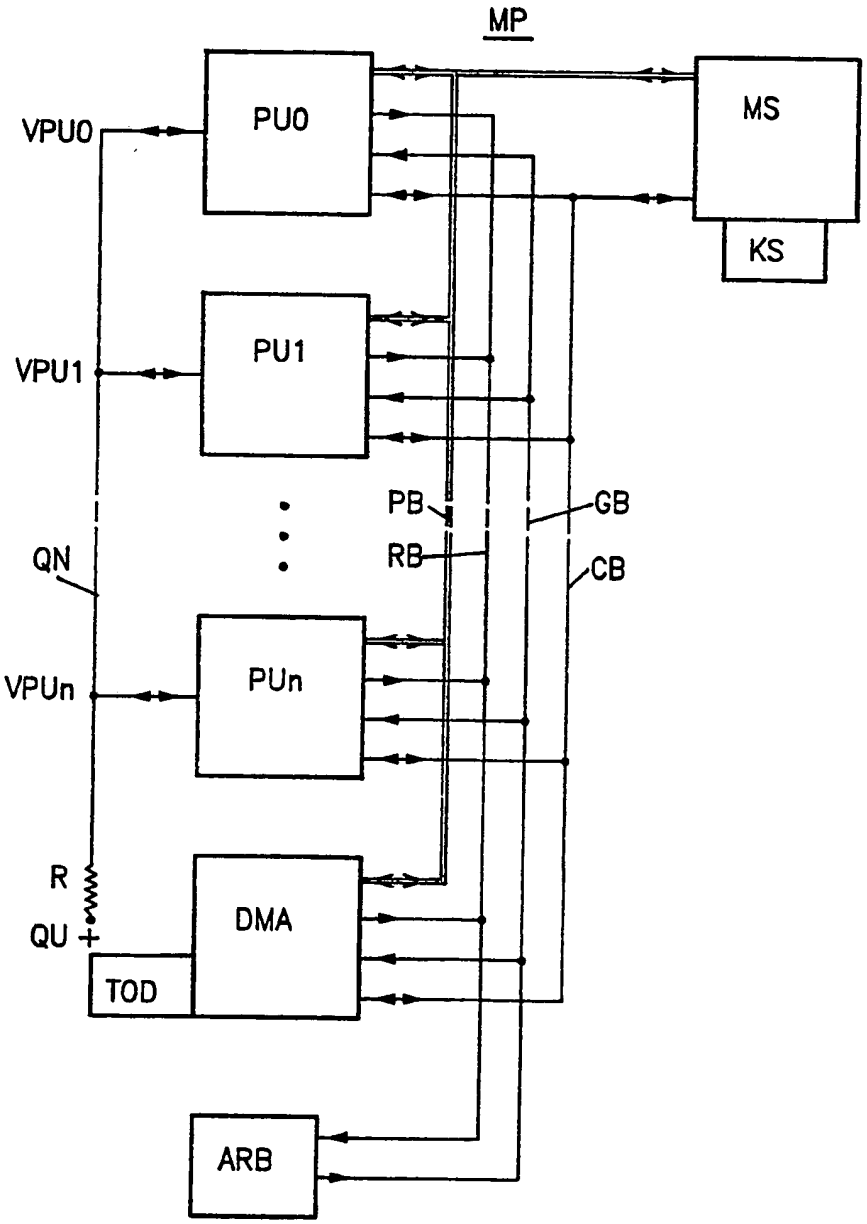

FIG. 1

Nummer: DE 43 07 139 A1
Int. Cl.[5]: G 06 F 13/36
Offenlegungstag: 8. September 1994

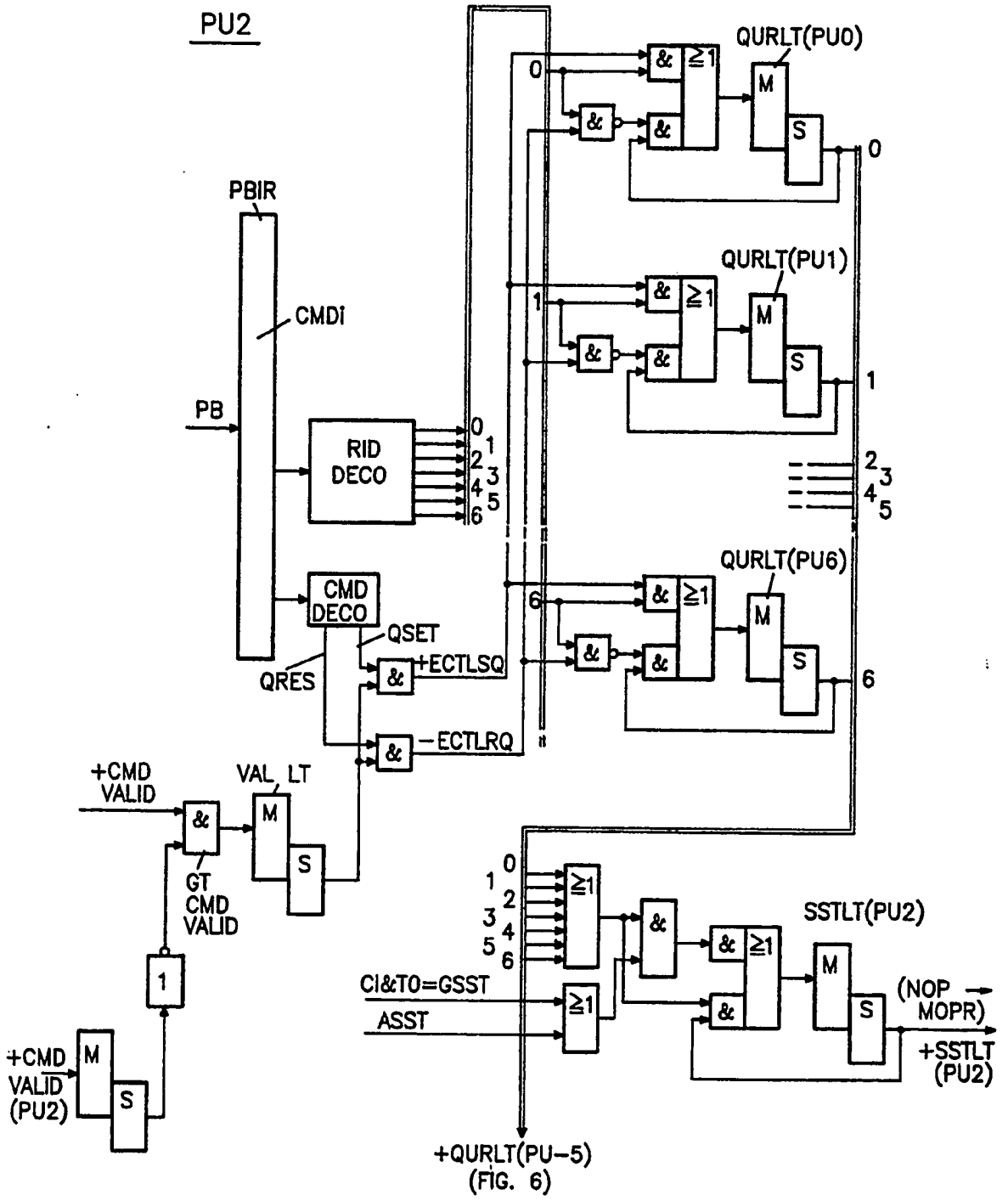

FIG. 2

Nummer: DE 43 07 139 A1
Int. Cl.[5]: G 06 F 13/36
Offenlegungstag: 8. September 1994

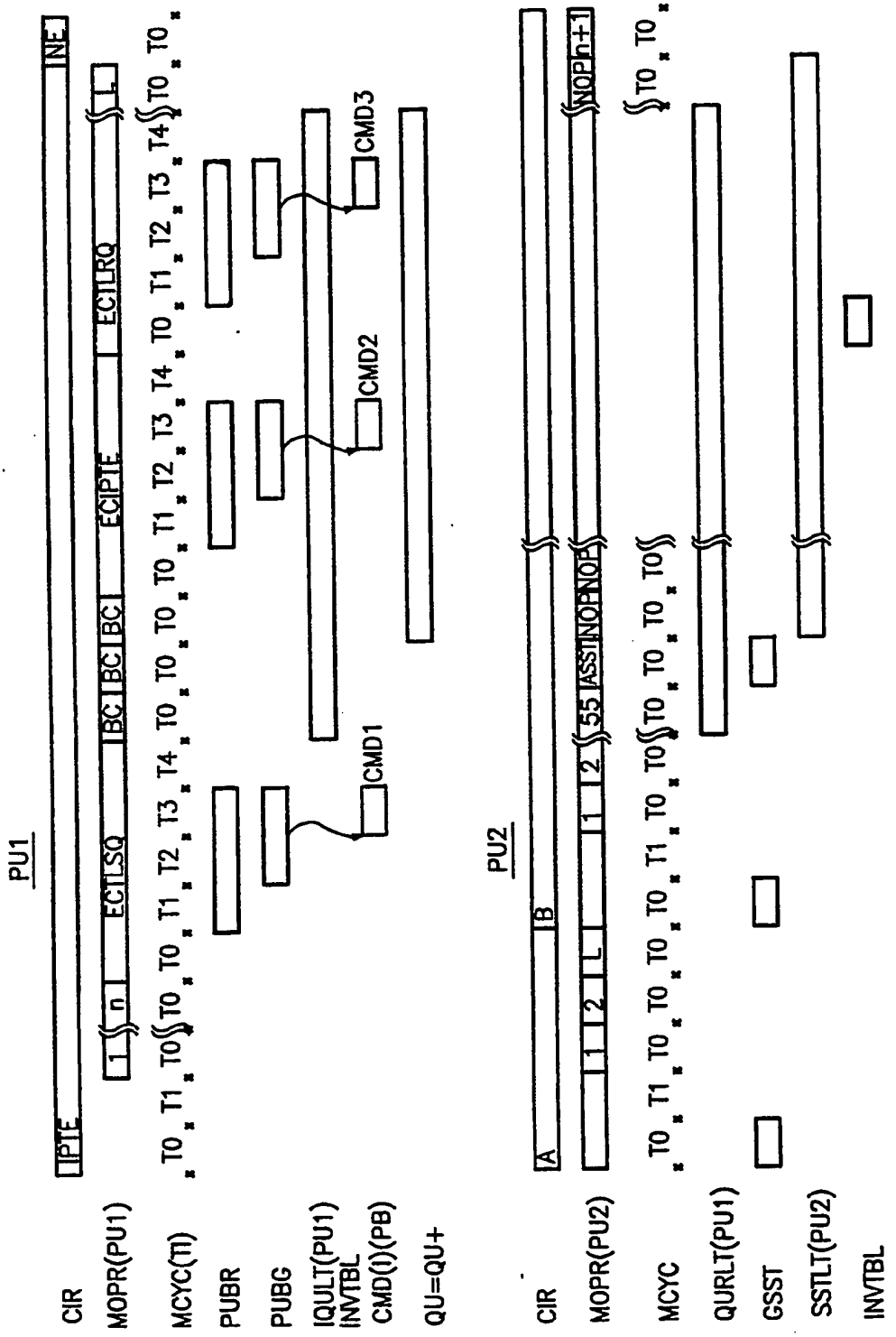

FIG. 3

Nummer: DE 43 07 139 A1
Int. Cl.[5]: G 06 F 13/38
Offenlegungstag: 8. September 1994

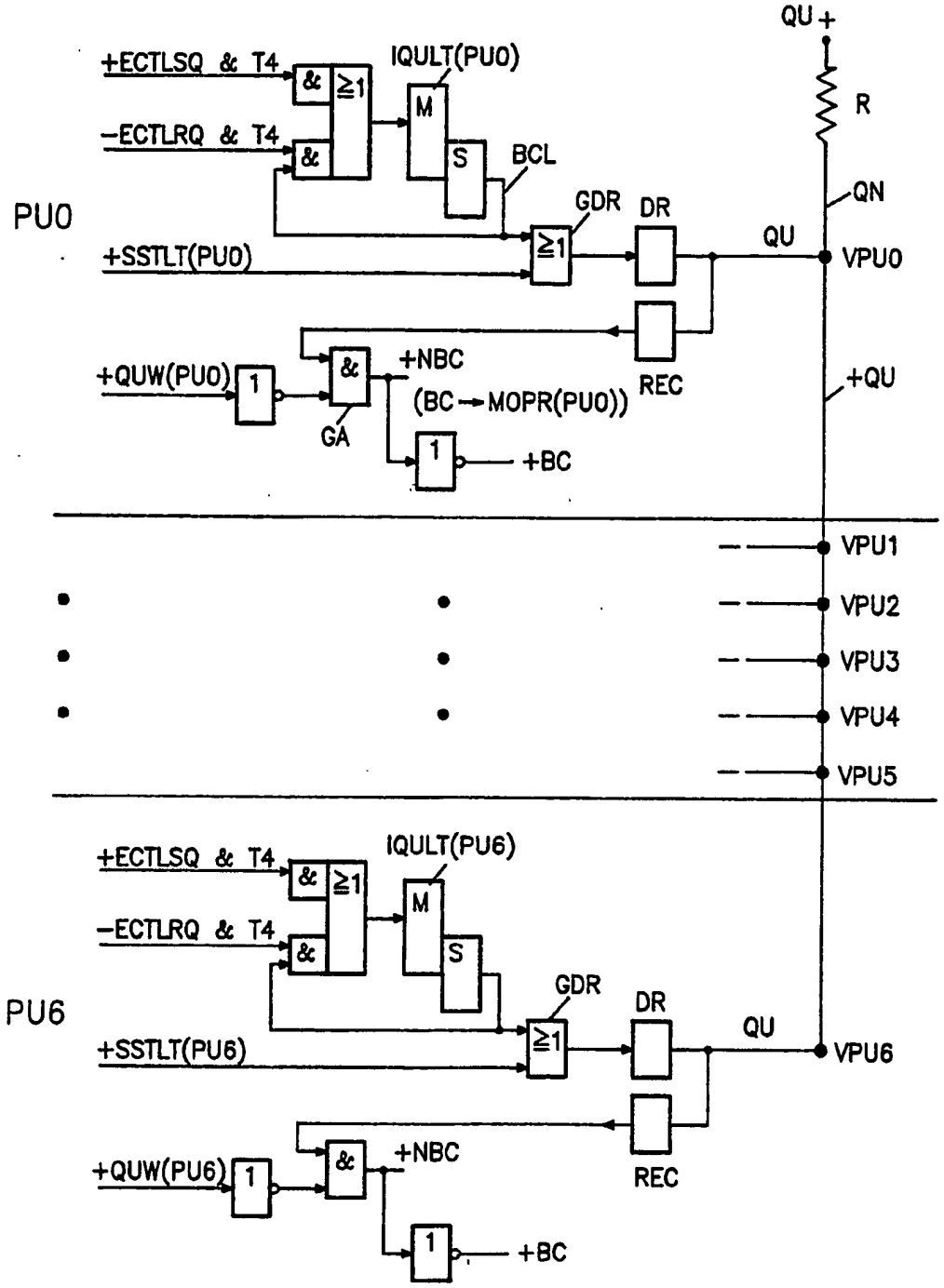

FIG. 4

Nummer: DE 43 07 139 A1
Int. Cl.[5]: G 06 F 13/36
Offenlegungstag: 8. September 1994

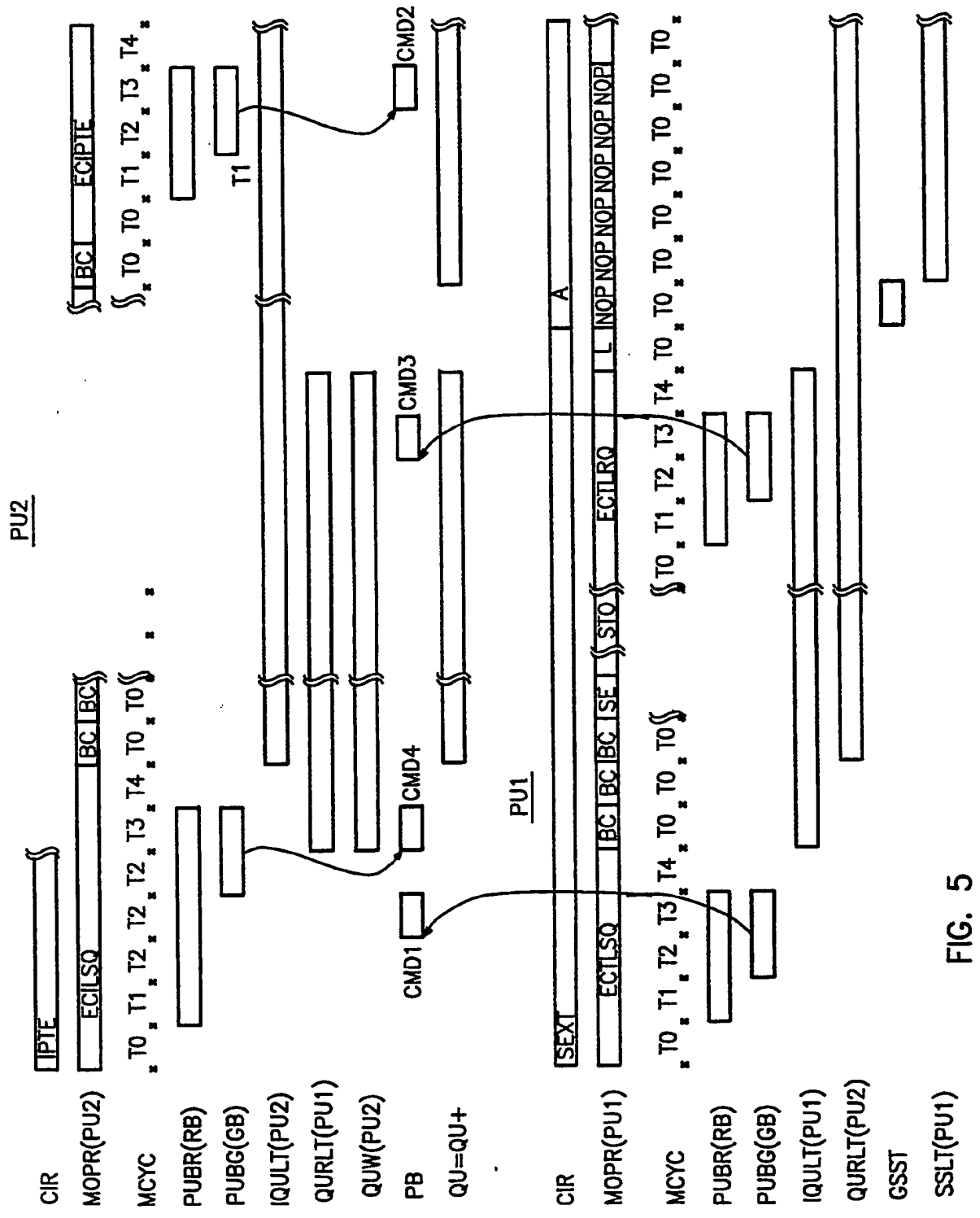

FIG. 5

Nummer: DE 43 07 139 A1
Int. Cl.[5]: G 06 F 13/36
Offenlegungstag: 8. September 1994

PU2

(FIG. 2)

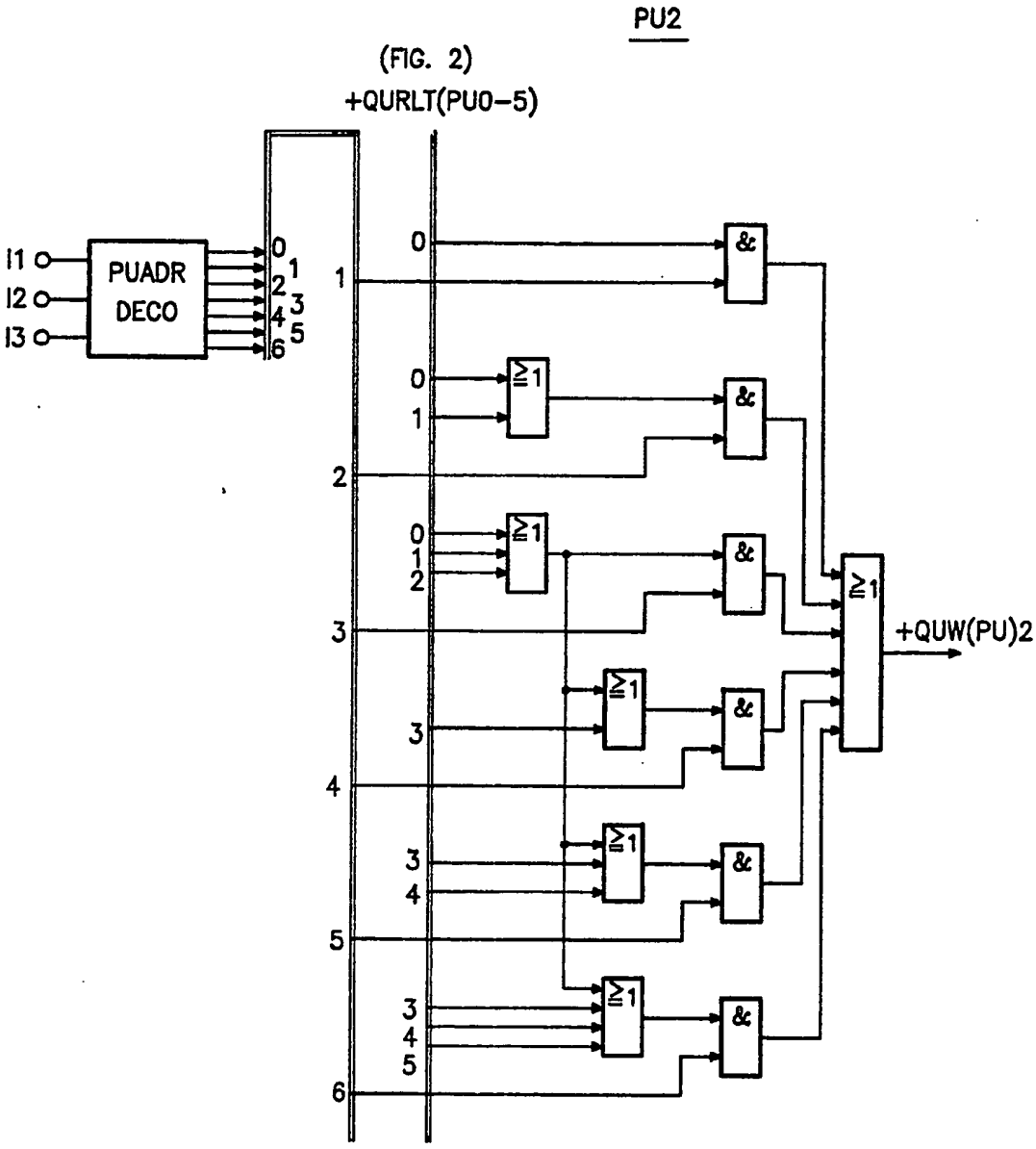

FIG. 6